IN THE CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Currently amended) A method for identifying a unique electronic mail message in a plurality of electronic mail messages extracted from an electronic mail messaging system, the method comprising:

retrieving from a first mailbox on the electronic mail messaging system a first message;

computing a first message tag ~~at least in part~~ by [[:1.]] concatenating a message sender of the first message and a message sender submission time of the first message into a first resulting string; ~~and 2. applying a hash algorithm to the first resulting string;~~

storing the first message in a message archive;

storing the first message tag in a single shared index file;

retrieving from a second mailbox on the electronic mail messaging system a second message, wherein the second mailbox is associated with a different electronic mail recipient than the first mailbox;

computing a second message tag ~~at least in part~~ by [[:1.]] concatenating a message sender of the second message and a message sender submission time of the second message into a second resulting string; ~~and 2. applying the hash algorithm to the second resulting string;~~ and

reviewing a list of message tags including the first message tag stored in the single shared index file, wherein:

i. in the event the second message tag matches the first message tag, determining the second message is a duplicate of the first message already stored in the message archive; and

ii. in the event the second message tag does not match any of the list of message tags including the first message tag, determining the second message is not a duplicate message already stored in the message archive and subsequently:

storing the second message in the message archive; and

storing the second message tag in the single shared index file.

2. (Cancelled)

3. (Currently amended) The method of claim [[1]]46, wherein applying [[a]]the hash algorithm to the message tag forms a uniform string, wherein the uniform string has a predetermined length.

4. (Original) The method of claim 3, wherein the hash algorithm is an MD5 hash algorithm.

5. (Canceled)

6. (Canceled)

7. (Original) The method of claim 1, wherein the index file is stored in a relational database system.

8. (Canceled)

9. (Canceled)

10. (Canceled)

11. (Canceled)

12. (Canceled)

13. (Canceled)

14. (Canceled)

15. (Currently amended) A system for identifying a unique electronic mail message, wherein the system is external to an electronic mail messaging system, the system comprising:

a processor configured to:

retrieve from a first mailbox on the electronic mail messaging system a first message;

compute a first message tag ~~at least in part~~ by [[:1.]] concatenating a message sender of the first message and a message sender submission time of the first message into a first resulting string; ~~and 2. applying a hash algorithm to the first resulting string;~~

store the first message in a message archive;

store the first message tag in a single shared index file;

retrieve from a second mailbox on the electronic mail messaging system a second message, wherein the second mailbox is associated with a different electronic mail recipient than the first mailbox;

compute a second message tag ~~at least in part~~ by [[:1.]] concatenating a message sender of the second message and a message sender submission time of the second message into a second resulting string; ~~and 2. applying the hash algorithm to the second resulting string;~~ and

review a list of message tags including the first message tag stored in the single shared index file, wherein:

i. in the event the second message tag matches the first message tag, determining the second message is a duplicate of the first message already stored in the message archive; and

ii. in the event the second message tag does not match any of the list of message tags including the first message tag, determining the second message is not a duplicate message already stored in the message archive and subsequently:

storing the second message in the message archive; and

storing the second message tag in the single shared index file; and

a memory configured to provide instructions to the processor.

16. (Canceled)

17. (Cancelled)

18. (Currently amended) The system of claim [[15]]47, wherein applying [[a]]the hash algorithm to the message tag forms a uniform string, wherein the uniform string has a predetermined length.

19. (Original) The system of claim 18, wherein the hash algorithm is an MD5 hash algorithm.

20. (Original) The system of claim 15, wherein the index file is stored in a relational database system.

21. (Canceled)

22. (Canceled)

23. (Canceled)

24. (Canceled)

25. (Canceled)

26. (Canceled)

27. (Canceled)

28. (Canceled)

29. (Canceled)

30. (Canceled)

31. (Canceled)

32. (Canceled)

33. (Canceled)

34. (Canceled)

35. (Canceled)

36. (Canceled)

37. (Currently amended) A system for externally archiving a plurality of electronic mail

messages selected from an electronic mail messaging system, the system comprising:

an archive server in communication with the electronic mail messaging system;

a duplicate checker in communication with the archive server; and

an archive message store in communication with the archive server,

wherein when the archive server receives a copy of a message from the electronic mail messaging system, a plurality of properties associated with the message are sent from the archive server to the duplicate checker,

wherein the duplicate checker computes a message tag for the message ~~using a subset of the properties, at least in part~~ by concatenating a message sender and a message sender submission time ~~and applying a hash algorithm to the resulting string,~~ and compares the computed message tag with a single shared index file, wherein the single shared index file stores message tags computed from respective messages properties of messages retrieved from a plurality of mailboxes associated with different electronic mail recipients,

wherein if the computed message tag matches an entry in the single shared index file, the duplicate checker indicates to the archive server that the message is a duplicate message with at least same said two message properties as another message already stored in the archive message store, otherwise, if the computed message tag does not match an entry in the single shared index file, the computed message tag is added to the single shared index file,

wherein if it is determined the message is not a duplicate message, the archive server stores the copy of the message in the archive message store;

wherein the copy of the message, if stored in the message archive, is archived for a mandated period of time.

38. (Cancelled)

39. (Currently amended) The system of claim [[37]]48, wherein applying [[a]]the hash algorithm to the message tag forms a uniform string, wherein the uniform string has a predetermined length.

40. (Original) The system of claim 39, wherein the hash algorithm is an MD5 hash algorithm.

41. (Original) The system of claim 37, wherein the archive server reads the message from a mailbox on the electronic mail messaging system.

42. (Canceled)

43. (Canceled)

44. (Canceled)

45. (Currently amended) A computer program product for identifying a unique electronic mail message in a plurality of electronic mail messages extracted from an electronic mail messaging system, the computer program product being embodied in a non-transitory computer readable storage medium and comprising computer instructions for:

retrieving from a first mailbox on the electronic mail messaging system a first message;

computing a first message tag ~~at least in part~~ by [[:1.]] concatenating a message sender of the first message and a message sender submission time of the first message into a first resulting string; ~~and 2. applying a hash algorithm to the first resulting string;~~

storing the first message in a message archive;

storing the first message tag in a single shared index file;

retrieving from a second mailbox on the electronic mail messaging system a second message, wherein the second mailbox is associated with a different electronic mail recipient than the first mailbox;

computing a second message tag ~~at least in part~~ by [[:1.]] concatenating a message sender of the second message and a message sender submission time of the second message into a second resulting string; ~~and 2. applying the hash algorithm to the second resulting string;~~ and

reviewing a list of message tags including the first message tag stored in the single shared index file, wherein:

i. in the event the second message tag matches the first message tag, determining the second message is a duplicate of the first message already stored in the message archive; and

ii. in the event the second message tag does not match any of the list of message tags including the first message tag, determining the second message is not a duplicate message already stored in the message archive and subsequently:

storing the second message in the message archive; and

storing the second message tag in the single shared index file.

46. (New) The method of claim 1, further comprising:

subsequent to computing the first message tag, replacing the first message tag with results of applying a hash algorithm to the first resulting string; and

subsequent to computing the second message tag, replacing the second message tag with results of applying the hash algorithm to the second resulting string.

47. (New) The system of claim 15, wherein the processor is further configured to:

subsequent to computing the first message tag, replacing the first message tag with results of applying a hash algorithm to the first resulting string; and

subsequent to computing the second message tag, replacing the second message tag with results of applying the hash algorithm to the second resulting string.

48. (New) The system of claim 37, wherein the duplicate checker is further configured to:

replacing the message tag with results of applying a hash algorithm to a resulting string of concatenating a message sender and a message sender submission time.

49. (New) The computer program product of claim 45, further comprising computer instructions for:

subsequent to computing the first message tag, replacing the first message tag with results of applying a hash algorithm to the first resulting string; and

subsequent to computing the second message tag, replacing the second message tag with results of applying the hash algorithm to the second resulting string.